Refine Search

Your wildcard search against 10000 terms has yielded the results below.

Your result set for the last L# is incomplete.

The probable cause is use of unlimited truncation. Revise your search strategy to use limited truncation.

Search Results -

Terms	Documents
L15 and distribut$ near3 (network or internet or www)	11

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Saturday, April 29, 2006 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L16	L15 and distribut$ near3 (network or internet or www)	11	L16
L15	hierarch$ near2 backup and retriev$	70	L15
L14	L11 and distribut$ near3 network	19	L14
L13	L11 and dbms near3 backup	1	L13
L12	L11 and (cells or nodes)	101	L12
L11	(hierarch$ or hierarchy or hierarchical) near2 (backup or back with up) and retriev$	155	L11
L10	711/162	2418	L10
L9	711/117	935	L9
L8	711/113	2003	L8

L7	711/112	2927	L7
L6	711.clas.	29314	L6
L5	707.clas.	34035	L5
L4	707/205	2162	L4
L3	707/204	2719	L3
L2	707/10	11910	L2
L1	707/1	7779	L1

END OF SEARCH HISTORY

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L16: Entry 9 of 11 File: USPT Nov 3, 1998

US-PAT-NO: 5832522
DOCUMENT-IDENTIFIER: US 5832522 A

TITLE: Data storage management for network interconnected processors

DATE-ISSUED: November 3, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Blickenstaff; Ronald L.	Niwot	CO		
Brant; Catherine Irlam	Boulder	CO		
Dodd; Paul David	Niwot	CO		
Kirchner; Anton H.	Nederland	CO		
Montez; Jennifer Kay	Thornton	CO		
Trede; Brian Eldred	Woodinville	WA		
Winter; Richard Allen	Longmont	CO		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Kodak Limited	Harrow			GB2	03

APPL-NO: 08/650114 [PALM]
DATE FILED: May 22, 1996

PARENT-CASE:
CROSS-REFERENCE TO RELATED APPLICATION This application is a divisional of a patent application entitled "Data Storage Management For Network Interconnected Processors," Ser. No. 08/201,658 filed Feb. 25, 1994, now U.S. Pat. No. 5,537,585.

INT-CL-ISSUED: [06] G06 F 17/30

US-CL-ISSUED: 707/204; 707/10, 707/205
US-CL-CURRENT: 707/204; 707/10, 707/205

FIELD-OF-CLASSIFICATION-SEARCH: 395/620, 395/621, 395/440, 395/441, 395/444, 395/800, 707/204, 707/205, 707/10
See application file for complete search history.

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5155835	October 1992	Belsan	395/441
☐	5218695	June 1993	Noveck et al.	395/600
☐	5276860	January 1994	Fortier et al.	395/575
☐	5276867	January 1994	Kenley et al.	395/600
☐	5313631	May 1994	Kao	395/600
☐	5367698	November 1994	Webber et al.	395/800
☐	5581724	December 1996	Belsan et al.	395/441

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	CLASS
WO 92/09035	May 1992	WO	

OTHER PUBLICATIONS

Proceedings of the Symposium On Mass Storage Systems, Symp. 11, Oct. 17, 1991 Institute of Electrical & Electronics Engineers, pp. 3-10, by Foster et al. Renaissance: Managing the Network Computer and its Storage Requirements.
Data Communications, vol. 22, No. 11, pp. 49-50, , by S. Salamone, "Migrating Data To Cheaper Storage".
Proceedings of the IEEE, vol. 63, No. 8, pp. 1166-1170, by C. Johnson, "The IBM 3850: A Mass Storage System with Disk Characteristics".
"Experience With File Migration" by R. D. Christman, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, reproduced by National Technical Information Service, U.S. Dept. of Commerce, Springfield, VA 22161, Oct. 1981.
"File Migration In The NCAR Mass Storage System" by E. Thanhardt & G. Harano, National Center for Atmospheric Research, Boulder, Colorado, Ninth IEEE Symposium on Mass Storage Systems, 1988.
"A Distributed Algorithm for Performance Improvement Through File Replication, File Migration, and Process Migration" by A. Had, IEEE Transactions on Software Engineering vol. 15, Issue 11, pp. 1459-1470, Nov. 1989.
"Development of Omniserver" by. D. A. Arneson, Control Data Corporation, Minneapolis, MN, 1990 IEEE, pp. 88-93.
"Architecture And Implementation of an On-line Data Archive and Distribution System", by. B. Bhasker, M. E. Van Steenberg, B.E. Jacobs, Hughes STX Corp., Lanham, MD, Proceedings Twelfth IEEE Symposium on Mass Storage Systems. Cat. No. 93CH3246-6, pp. 177-182, Apr. 1993.
"Potential Benefits of File Migration in a Heterogeneous Distributed File Systems" by R. T. Hurley, S.A. Yeap, J. W. Wong, J.P. Black, Proceedings ICCI '93, Fifth International Conference on Computing and Information, Cat. No. 93Th0563-7, pp. 123-127, May 1993.

ART-UNIT: 271

PRIMARY-EXAMINER: Lintz; Paul R.

ATTY-AGENT-FIRM: Duft, Graziano & Forest, P.C.

ABSTRACT:

The data storage system is connected to a local area network and includes a storage server that on a demand basis and/or on a periodically scheduled basis audits the activity on each volume of each data storage device that is connected to the network. Low priority data files are migrated via the network and the storage server to backend data storage media, and the directory resident in the data storage device is updated with a placeholder entry to indicate that this data file has been migrated to backend storage. When the processor requests this data file, the placeholder entry enables the storage server to recall the requested data file to the data storage device from which it originated.

25 Claims, 13 Drawing figures

Previous Doc Next Doc Go to Doc#

L16: Entry 9 of 11 File: USPT Nov 3, 1998

DOCUMENT-IDENTIFIER: US 5832522 A
TITLE: Data storage management for network interconnected processors

Brief Summary Text (6):
In addition, the retrieval of archived data files is cumbersome since the identification of archived data files is typically expunged from the file server and listed in a separate archived files directory. Thus, the file server must first scan the file server directory, then the archived files directory in response to a host processor request for an archived data file. This recursive search process is wasteful of processing resources.

Brief Summary Text (10):
The above-described problems are solved and a technical advance achieved in the field by the data storage management system of the present invention. The data storage management system is connected to the network and provides a hierarchical data storage capability to migrate lower priority data files from the data storage subsystems that are connected to the network to backend less expensive data storage media, such as optical disks or magnetic tape. A data storage management capability is also included to provide automated disaster recovery data backup and data space management capability. In particular, a placeholder entry is inserted into the directory entry in the managed file server volume for each migrated data file. The placeholder entry both indicates the migrated status of the data file and provides a pointer to enable the requesting processor to efficiently locate and retrieve the requested data file.

Brief Summary Text (11):
The data storage management system implements a virtual data storage system, comprising a plurality of virtual file systems, for the processors that are connected to the network. The virtual data storage system consists of a first section that comprises a plurality of data storage subsystems, each consisting of file servers and their associated data storage devices, which are connected to the network and serve the processors. A second section of the virtual data storage system comprises the storage server, consisting of a storage server processor and at least one layer of hierarchically arranged data storage devices, that provides backend data storage space. The storage server processor interfaces to software components stored in each processor and file server that is connected to the network. The storage server, on a demand basis and/or on a periodically scheduled basis, audits the activity on each volume of each data storage device that is connected to the network. Data files that are of lower priority are migrated via the network and the storage server to backend data storage media. The data file directory resident in the data storage device that originally contained this data file is updated with a placeholder entry in the directory to indicate that this data file has been migrated to backend data storage. Therefore, when a processor requests this data file, the placeholder entry is retrieved from the directory and the storage server is notified that the requested data file has been migrated to backend storage and must be recalled to the data storage device from which it originated. The storage server automatically retrieves the requested data file using information stored in the placeholder entry and transmits the retrieved data file to the data storage device from whence it originally came. The storage server, backend data storage and processor resident software modules create a virtual

storage capacity for each of the data storage devices in a manner that is transparent to both the processor and the user. Each virtual volume in this system can be expanded in extent in a seamless manner to match the needs of the processor by using low cost mass storage devices.

Brief Summary Text (12):
In operation, the storage server monitors the amount of available data storage space on each of the volumes (network volumes) on each of the data storage devices to ensure that adequate data storage space is available to the processors on a continuing basis. When the available data storage space drops below a predetermined threshold, the storage server reviews the activity levels of the various data files that are stored therein and automatically migrates the lower priority data files to the backend data storage as described above. Furthermore, the backend data storage is similarly managed with the lower priority data files being migrated from layer to layer within the multi-layer hierarchical data storage as a function of their activity level, content and the amount of available data storage space on these various layers. Therefore, each layer of the hierarchical storage is populated by data files whose usage pattern and priority is appropriate to that layer or type of media. The data storage devices can be viewed as comprising a first layer of this data storage hierarchy while a backend disk drive or disk drive array can be a second layer of this data storage hierarchy. Successive layers of this hierarchy of data storage devices can incorporate optical disks, and/or magnetic tape, and/or automated media storage and retrieval libraries, and/or manual media storage and retrieval libraries.

Detailed Description Text (4):
In addition to the processors 21, 22 and the file servers 41-43, the data storage management system of the present invention includes the data storage management apparatus connected to the local area network 1. This data storage management apparatus comprises a storage server 50 that is connected to the local area network 1. A storage server processor 51 serves to interface the local area network 1 with the backend data storage devices 61-65 (FIG. 4) that constitute the secondary storage 52. The backend data storage devices 61-65, in combination with the file servers 41-43 comprise a hierarchical data storage system. The backend data storage devices 61-65 typically include at least one layer of data storage that is less costly than the dedicated data storage devices 31-33 of the file servers 41-43 to provide a more cost-effective data storage capacity for the processors 21, 22. The data storage management system implements a virtual data storage space for the processors 21, 22 that are connected to the local area network 1. The virtual data storage space consists of a first section A that comprises a primary data storage device 31 that is connected to the network 1 and used by processors 21, 22. A second section B of the virtual memory comprises the secondary storage 52 managed by the storage server processor 51. The secondary storage 52 provides additional data storage capacity for each of the primary data storage devices 31-33, represented on FIG. 1 as the virtual devices 31S-33S attached in phantom to the primary data storage devices 31-33 of the file servers 41-43. Processor 21 is thereby presented with the image of a greater capacity data storage device 31 than is connected to the file server 41. The storage server 51 interfaces to software components stored in each processor 21, 22 and file server 41-43 that is connected to the local area network 1. The storage server processor 51, on a demand basis and/or on a periodically scheduled basis, audits the activity on each volume of each data storage device 31-33 of the file servers 41-43 that are connected to the network 1. Data files that are of lower priority are migrated via the network 1 and the storage server processor 51 to backend data storage media of the secondary storage 52. The data file directory resident in the file server 41 that originally contained this data file is updated with a placeholder entry in the directory to indicate that this data file has been migrated to backend data storage. Therefore, when the processor 21 requests this data file, the placeholder entry is retrieved from the directory and the storage server processor 51 is notified that the requested data file has been migrated to backend storage and must be recalled to

the file server 41 from which it originated. In the case of a processor 21, 22 and 42 that interfaces to a user, the storage server 50 may provide the user with a notification where necessary that a time delay may be noted in accessing the requested data file. The storage server processor 51 automatically retrieves the requested data file and transmits it to the data storage device 31 from whence it originally came. The storage server processor 51, secondary storage 52 and processor resident software modules create a virtual storage capacity for each of the file servers 41-43 in a manner that is transparent to both the processor 21, 22 and the user. Each virtual volume in this system can be expanded in extent in seamless manner to match the needs of the processors 21, 22 by using low cost mass storage devices to implement the secondary storage 52.

Detailed Description Text (11):
As illustrated in FIG. 3, the secondary storage 52 is divided into at least one and more likely a plurality of layers 311-313, generally as a function of the media used to implement the data storage devices 61-65. In particular, the second layer 311 of the hierarchical data storage, which is the first layer of the secondary storage 52, can be implemented by high speed magnetic storage devices 61. Such devices include disk drives and disk drive arrays. The third layer 312 of the hierarchical data storage, which is the second layer of the secondary storage 52, can be implemented by optical storage devices 62. Such devices include optical disk drives and robotic media storage and retrieval library systems. The fourth layer 313 of the hierarchical data storage, which is the third layer of the secondary storage 52, can be implemented by slow speed magnetic storage devices 63. Such devices include magnetic tape drives and robotic media storage and retrieval library systems. An additional layer 314 of the hierarchical data storage can be implemented by the use of a "shelf layer", which can be implemented by manual storage of media. This disclosed hierarchy is simply illustrative of the data storage management concept and the number, order and implementation of the various layers can differ from that disclosed herein.

Detailed Description Text (14):
As data files are transmitted to the storage server 51 for migration to secondary storage 52, they are automatically protected from loss in several ways. The data storage devices 61 in the first layer 311 of the second section of the virtual data storage system are typically protected by the use of shadow copies, wherein each data storage device 61 and its contents are replicated by another data storage device 65 and its contents. In addition, as data files are migrated to the storage server 51 for retention, they are packaged into large blocks of data called transfer units. The transfer units are backed up via a backup drive 71 on to a separate backup media 72, such as high density magnetic tape media. Multiple copies of this backup media 72 may be created to provide both off-site and on-site copies for data security. A backup media rotation scheme can be implemented to rotate the backup media between a plurality of locations, typically between an on-site and an off-site location to protect against any physical disasters, such as fire. When the lowest layer 313 of the second section of the virtual data storage space becomes nearly full, the data storage devices 63 that comprise this layer are reviewed to identify the lowest priority transfer units contained thereon. These identified transfer units are deleted from this layer and the secondary storage directories are updated to indicate that the data files contained in these deleted transfer units have been "relocated" to the shelf layer 314. No physical movement of the transfer units or the data files contained therein takes place. The relocation is virtual, since the data files are presently stored on backup media 72 that was created when these identified data files were initially migrated to the first layer of the secondary storage. The placeholder entry for each of the data files contained in the deleted transfer units is not updated, since the data files are still accessible within the data storage system. The secondary storage directories are updated to note that the data files are presently stored on the shelf layer 314 and the identity of the media element 72 that contains this data file is added to the directory entry for this data file. This shelf storage concept is very

convenient for temporary overflow situations where free space is required at the lowest layer 313 of the hierarchy but the user has not procured additional data storage devices 63. Where the user subsequently does expand the data storage capacity of this layer, the overflowed data can be automatically retrieved from the shelf storage and placed in the additional data storage space.

Detailed Description Text (15):
When a processor 21 requests access to a data file that is stored in the shelf layer 314, the storage server 51 retrieves the physical storage location data from the secondary storage directory associated with the requested data file. This data includes an identification of the media element 72 that contains the requested data file. The physical location of this media element 72 is dependent on the data read/write activity and configuration of the system. It is not unusual for the identified media element 72 to be mounted on the backup drive 71 that performs the data file backup function. If so, the data file is retrieved from this backup drive 71. If the media element 72 has been removed from the backup drive 71, an operator must retrieve the removed media element 72 and mount this media element on a drive 71 to enable the storage server 51 to recall the requested data file from the media element 72 and transmit the data file to the file server 31 used by the requesting processor 21. The retrieved media element 72 can be mounted on the backup drive 71 or a separate drive can optionally be provided for this purpose to enable the storage server 51 to continually backup data files as they are migrated to secondary storage 52. Thus, the backup media 72 serves two purposes: backup of data files, and shelf layer 314 of storage in the data storage hierarchy.

Detailed Description Text (17):
When data files have not been utilized for an extended period of time, they should be removed from the virtual data storage system and placed in another managed data storage system that does not utilize the more expensive automatic resources of the virtual data storage system. It is advantageous to track these retired data files in the event that they need to be retrieved. The retirement layer 315 performs this function. When a data file is retired, it no longer is part of the virtual data storage system and its placeholder entry is deleted from the primary storage directory. In addition, the identification of the data file and any other properties that were recorded in the secondary storage directory are saved and placed in a separate retirement directory. The retired file's placeholder entry, secondary storage directory entry and backup directory entry are deleted.

Detailed Description Text (20):
The data management system software of the present invention manages the flow of data files throughout the system. The block diagram of FIG. 10 illustrates a conceptual client-server view of the network and the data management system software. The data communication link 11 of the local area network 1 is illustrated having the storage server processor 51 and three file systems 41-43 attached thereto. The storage server processor 51 includes the network operating system 111 as well as the data storage management system software consisting of various media and device management user interfaces 112 and control and services software 113. Each file server 41-43 includes a storage server agent 121-123 and any processor of the network can include and run an administrative user interface 131. The control and services software 113 looks at the system as a set of clients that are connected to the network 1 and which require services from the storage server 50. Each file server 41-43 communicates with the storage server processor 51 via the resident storage server agent software 121-123. Thus, the data management system software is distributed throughout the network and serves to transparently integrate all the elements connected to the network into the data storage hierarchy.

Detailed Description Text (48):
This file system separates the logical allocation of data storage from the physical storage allocation, with the logical allocation for all layers of the data storage

hierarchy being the same since the data file remains in its unique transfer unit. One significant advantage of this system is that when transfer units are migrated from layer to layer in the hierarchy or placed on a backup media, only the relationship between transfer unit identification and media object need be updated to reflect the new media on which this transfer unit is stored. Furthermore, the data file retains its relationship to the transfer unit in the backup system, and the backup media simply provides a redundant media object for the same transfer unit identification. The transfer unit is then written into the first layer 311 of the secondary storage 52. This procedure is used to relocate transfer units from one layer in the data storage hierarchy to the next lower layer in the data storage hierarchy. The block diagram of FIG. 11 illustrates the nested nature of the transfer units. Thus, the transfer unit of data files from the primary storage represents a data block of a first extent. The second layer transfer unit, assembled to relocate data files from the first layer of the hierarchical data storage to the second layer, can be composed of a plurality of first layer transfer units. Similarly, this process can be applied to successive layers of the data storage hierarchy. FIG. 11 illustrates the resultant stream of data that is written on to the lowest layer of the data storage hierarchy for a three layer secondary storage, consisting of a plurality of sequentially ordered second layer transfer units, each of which is comprised of a plurality of first layer transfer units.

Detailed Description Text (51):
The number and configuration of the layers of the hierarchy can be dynamically altered to suit the needs of the user. Additional layers can be added to the hierarchy or deleted therefrom. In addition, data storage capacity can be added or deleted from any selected layer of the hierarchy by the inclusion or exclusion of data storage devices from that selected layer. The data storage management system automatically adapts to such modifications of the hierarchy in a manner that ensures maximum performance and reliability. The shelf layer that is implemented by the backup drive 71 and the mountable backup data storage element 72 can provide an overflow capacity for the first layer 311 of the secondary storage 52 if no additional layers are provided, or for the lowest layer 313 if multiple layers are provided. Thus, when there is no longer any available data storage space on the lowest layer of the hierarchy, transfer units or media units are deleted from this layer. If additional data storage capacity in the form of additional data storage devices are added to this layer, or alternatively, an additional layer of media is provided below the previously lowest layer of media, the deleted transfer or media units can be returned to the hierarchy from the backup mountable data storage elements 72. This is accomplished by the storage server 51 noting the presence of newly added available data storage space on the lowest layer of the hierarchy and previously deleted transfer or media units. The storage server 51 accesses the media object directory to identify the location of the deleted data and retrieve this data from an identified backup mountable data storage element 72, which is mounted on backup drive 71. This retrieved data is then written on to the newly added media in available data storage space. This process is also activated if a data storage device is removed from a layer of the media or added to a layer of the media. If this media modification occurs in any but the lowest layer, the deleted transfer units or media objects are retrieved from the backup mountable data storage element 72 and stored on the same layer as they originally were stored unless insufficient space is available on that layer, in which case they are stored on the media level immediately below the level on which the data storage device was removed.

Detailed Description Text (53):
As illustrated in flow diagram form in FIG. 8 and with reference to the system architecture in FIG. 7, a data file recall operates in substantially the reverse direction of data file migration. As noted above, the data files that are written to the migration volumes 61 and shadow volumes 65 have their physical storage location identification written into a secondary storage directory 531 in the file server 41. The placeholder entry in directory 511 on the file server 41 points to

this secondary storage directory entry. Thus, the processor 21 at step 801 requests access to this migrated data file and this request is intercepted at step 802 by a trap or interface 711 in the file server 41. The trap can utilize hooks in the file system 41 to cause a branch in processing to the storage server agent 121 or a call back routine can be implemented that allows the storage server agent 121 to register with the file system 41 and be called when the data file request is received from the processor 21. In either case, the trapped request is forwarded to storage server agent 121 to determine whether the requested data file is migrated to secondary storage 52. This is accomplished by storage server agent 121 at step 803 reading directory 511 to determine the location of the requested data file. If a placeholder entry is not found stored in directory 511 at step 805, control is returned to the file server 41 at step 806 to enable the file server 41 to read the directory entry that is stored in directory 511 for the requested data file. The data stored in this directory entry enables the file server 41 to retrieve the requested data file from the data storage device 31 on which the requested data file resides. If at step 805, storage server agent 121 determines, via the presence of a placeholder entry, that the requested data file has been migrated to secondary storage 52, storage server agent 121 at step 807 creates a data file recall request and transmits this request together with the direct access secondary storage pointer key stored in the placeholder entry via network 1 to storage server 50. At step 808, operations kernel 501 uses systems services 505 which uses the pointer key to directly retrieve the entry in secondary storage directory 531. This identified entry in the secondary storage directory 531 contains the address in the migration volume that contains the requested data file. The address consists of the transfer unit identification and position of the data file in the transfer unit. The device manager 504 uses the data file address information to recall the requested data file from the data storage device on which it is stored. This data storage device can be at any level in the hierarchy, as a function of the activity level of the data file. Device manager 504 reads the data file from the storage location in the data storage device identified in the secondary storage directory 531 and places the retrieved data file on the network 1 for transmission to the file server 41 and volume 31 that originally contained the requested data file. Systems services 505 of operations kernel 501 then updates the secondary storage directory 531 and the directory 511 to indicate that the data file has been recalled to the network volume. At step 811, control is returned to file server 41, which reads directory 511 to locate the requested data file. The directory 511 now contains information that indicates the present location of this recalled data file on data storage device 31. The processor 21 can then directly access the recalled data file via the file server 41.

Detailed Description Text (57):
In addition, the secondary storage directory 531, since it is distributed on network volumes, is backed up on to the primary storage backup media as noted above. This metadata can also be optionally replicated into a data storage device of the secondary storage or backed up on to the backup media 72.

CLAIMS:

3. The system of claim 1 further comprising:

means, located in each of said plurality of file servers, for intercepting a call at a selected file server to a data file that has been stored in said file server;

means, responsive to said data written in said directory means indicating that said requested data file has been migrated to said secondary storage means, for recalling said requested data file from said secondary storage means to said file server, comprising:

means for reading said data stored in said directory means to identify a physical data storage location in said storing means that contains data which identifies a

locus in said secondary storage means of said requested migrated data file,

means for retrieving said data stored in said identified physical storage location in said storing means, and

means, responsive to said retrieved data, for transmitting said requested migrated data file from said locus in said secondary storage means to said selected file server.

16. The method of claim 14 further comprising:

intercepting, in each of said plurality of file servers, a call at a selected file server to a data file that has been stored in said file server;

recalling, in response to said data written in said directory indicating that said requested data file has been migrated to said secondary storage system, said requested data file from said secondary storage system to said file server, comprising the steps of:

reading said data stored in said directory to identify a physical data storage location in said memory that contains data which identifies a locus in said secondary storage system of said requested migrated data file,

retrieving said data stored in said identified physical storage location in said memory, and

transmitting, in response to said retrieved data, said requested migrated data file from said locus in said secondary storage system to said selected file server.

Previous Doc Next Doc Go to Doc#